Remarks

Status of the Claims

Claims 1-3, 5-17, 19-25, 27-49, and 51-67 were pending in the application. All claims were finally rejected in the Office Action mailed August 11, 2006. By this paper, claims 1, 3, 5, 17, 25, 27, 41, 49, 51, 65, 66, and 67 have been amended. In view of the amendments and following remarks, reconsideration of the claims is respectfully requested.

Interview

Applicant would like to thank the Examiner and her supervisor for the courtesy of the interview on October 7, 2006. During the interview, Applicant's representative discussed the rejections under Sections 101 and 103.

Specifically, with respect to the 101 rejection, Applicant's representative argued that the rejected claims are statutory "article of manufacture" claims under the *Interim Guidelines* and *In re Beuregard*. With regard to the 103 rejections, Applicant's representative argued that none of the cited references teach or suggest specific data structures, but, rather, merely disclose "program schedule information." In addition, Applicant's representative argued that the cited references do not teach or suggest the claimed encapsulation of attribute data, program code, and graphical data for a particular television program, which allows the claimed data/code to be transmitted as a unit between interactive television systems. Agreement appears to have been reached that the prior art of record does not include these feature, but Applicant understands that the Examiner will need to update her search.

Section 101 Rejection

Claims 1-16 were rejected under 35 U.S.C. 101 because the claimed invention is allegedly directed to non-statutory subject matter. In Applicant's amendment dated June 13, 2006, claims 1-16 were amended to recite a computer-readable medium. However, the Examiner maintains that claims 1-16 still "do not impart functionality." Applicant respectfully disagrees.

According to the Interim Guidelines,

> a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Interim Guidelines, Annex IV (page 52).

There is no question that claim 1 recites functionality. As amended, claim 1 recites a computer-readable medium including:

> program code for a plurality of ***user-selectable actions performable by the interactive television system*** in connection with the television program, the program code for ***causing the interactive television system to carry the respective actions***; and
>
> graphical data for displaying a visual indicator in a graphical user interface to facilitate user interaction with the PIO, wherein the attribute data, program code, and graphical data for the particular program are transmittable as a unit from one interactive television system to another in response to the encapsulating PIO being sent between the interactive television systems.

Applicant respectfully submits that these limitations define "structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized." In

particular, the program code causes an interactive television system ("hardware components") to carry out user-selected actions (*e.g.*, recording, reminding, displaying information) in connection with a particular television program. Thus, the program code contained within a PIO causes a functional change within a computer. In addition, the visual indicator of the PIO within the graphical user interface ("computer software") allows a user to interact with the PIO, *e.g.*, selecting actions to be performed by the interactive television system.

In the interview of November 7, 2006, the Examiner inquired about the statutory class of claim 1. Claim 1 has been amended to recite an article of manufacture including a computer-readable medium. This is similar to the language at issue in *In re Beauregard*, Appeal No. 95-1054 (Fed. Cir., filed November 15, 1994), which was found to recite statutory subject matter.

The language of claim 1 is consistent with *In re Lowry,* 32 F.3d 1579, 32 U.S.P.Q.2d 1031 (Fed. Cir. 1994), in which the Federal Circuit held that a data structure in a computer memory was statutory subject matter. A portion of a representative claim from Lowry is as follows:

> 1. A memory for storing data for access by an application program being executed on a data processing system, comprising:
>
> a data structure stored in said memory, said data structure including information resident in a database used by said application program and including:
>
> a plurality of attribute data objects stored in said memory, each of said attribute data objects containing different information from said database;
>
> a single holder attribute data object for each of said attribute data objections, each of said holder attribute data objects being. . . ; . . .

an apex data object stored in said memory and having no being-held relationship with any of said attribute data objects, however, at least one of said attribute data objects having a being-held relationship with said apex data object.

The Court characterized this claim as imposing a physical organization on the data since the data structures were specific electrical or magnetic structural elements in a memory. By claiming "a memory for storing data" and specifying that the data structure was "for access by an application program being executed on a data processing system" in the preamble, Lowry recited a computer-readable memory that could direct a computer to function in a particular manner.

Accordingly, Applicant respectfully submits that the claimed program interface object in a computer-readable medium satisfies the definition of "functional descriptive material" as set forth in the Interim Guidelines and is therefore statutory.

Section 112 Rejections

Claims 41-64 were rejected under 35 U.S.C. 112, first paragraph, as allegedly failing to comply with the written description requirement. Specifically, the Office Action objected to the phrase "discrete container" as not being found in the specification.

Applicant respectfully submits that a person of ordinary skill in the art would understand that a "data structure" is a type of "container," *i.e.*, a container of data. Nevertheless, to advance prosecution of the application, Applicant has amended claim 41 to recite a "discrete ***data structure***" rather than a "discrete container."

Section 103 Rejections

Claims 1-3, 5, 7-11, 14, 15, 17, 19, 24, 25, 27, 29-33, 36-38, 41-43, 48, 49, 51, 53-57, 60-62, and 66 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima et a. ("Niijima"). Claims 6, 28, and 52 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan and in view of Niijima and further in view of Maryka et al. ("Maryka"). Claims 12, 34, and 58 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and further in view of Hassell et al. ("Hassell"). Claims 13, 35, and 59 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and further in view of Ellis et al. ("Ellis"). Claims 16, 20, 39, 44, and 63 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and further in view of Lawler et al. ("Lawler"). Claims 21-23, and 45-47 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and further in view of Knudson et al. ("Knudson"). Claims 40 and 64 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and further in view of Young et al. ("Young"). Claims 65 and 67 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sullivan in view of Niijima and Kamen et al. ("Kamen"). These rejections are respectfully traversed. Applicant respectfully submits that each of the pending claims, as amended, is patentably distinct from the cited references, individually and collectively.

Claims 1, 17, 41, and 65-67

Independent claims 1, 17, 41, and 65-67 have been variously amended to recite that a program interface object (PIO) comprises a ***separate data structure*** within a ***memory*** of an interactive television system for ***encapsulating***:

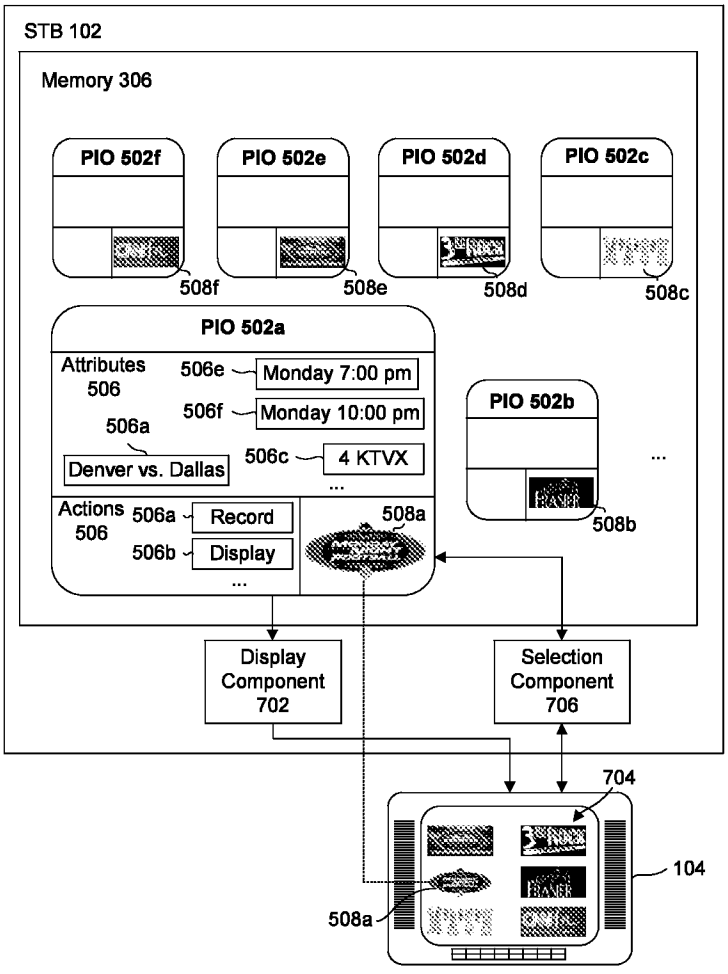
> ***attribute data*** for a plurality of attributes carrying information about the television program;
>
> ***program code*** for a plurality of user-selectable actions performable by the interactive television system in connection with the television program, the program code for causing the interactive television system to carry out the respective actions; and
>
> ***graphical data*** for displaying a visual indicator in a graphical user interface to facilitate user interaction with the PIO, wherein the ***attribute data, program code, and graphical data*** associated with the particular program ***are transmittable as a unit*** from one interactive television system to another ***in response to the encapsulating PIO being sent between the interactive television systems***.

Support for these amendments may be found, for example, in Figures 5 and 15 (and their accompanying text), as well as pages 1 and 2 of the specification.

Therefore, a PIO, as claimed, is a unique type of data structure for representing a particular television program within a memory of an interactive television system. During the interview, the Examiner's supervisor suggested that because the prior art teaches distribution of data ("program schedule information") in digital form, that this might suggest the particular **structure** of data recited in claim 1. Applicant respectfully disagrees. This would be akin to arguing that because steel can be molded into various shapes, any claimed mechanical device made out of steel is automatically rendered obvious. As argued more fully below, the prior art merely discloses "program schedule information." The Office Action appears to be using the prior art for the raw material of

"program schedule information," and then relying on the Applicant's own disclosure for the organization of that data, which is impermissible hindsight reconstruction.

As illustrated below in Figure 7 of the present application, each PIO exists as a discrete object or data structure including all of the attributes, actions, and visual/audible indicators needed for interacting with the corresponding television program. For instance, the PIO 502a includes a number of attributes, such as the name of the program ("Dallas v. Denver"), the starting time ("Monday, 7:00 p.m."), etc. In addition, the PIO 502a includes a number of actions ("Record"), which are represented within the PIO 502a as program code for carrying out the actions.



1. The cited references do not disclose anything about underlying ***data structures.***

The term "data structure" is well understood to those of ordinary skill in the art of computer science/programming:

> (1) "In computer science, a data structure is a way of storing data in a computer so that it can be used efficiently." (www.wikipedia.org).
>
> (2) "In programming, the term data structure refers to a scheme for organizing related pieces of information. The basic types of data structures include:
>
> files
> lists
> arrays
> records
> trees
> tables
>
> Each of these basic structures has many variations and allows different operations to be performed on the data." (www.webopedia.com).

The Office Action refers to various elements of the cited references as "PIOs." None of them, however, are "data structures" in the sense discussed above. Moreover, none of the references disclose or even suggest a single data structure that contains or encapsulates (1) ***attribute data***, (2) ***program code for user-selectable actions,*** and (3) ***graphical data for a visual indicator.***

For example, with regard to Sullivan, the Office Action refers to Figures 3A and 3B, as well as column 7, lines 37-67, and column 8, lines 1, 29-56, for a "PIO comprising a discrete data structure." However, a closer look at Sullivan reveals that the reference does not shed any light on the underlying data structures used to implement Sullivan's electronic program guide (EPG). For instance, Figure 3A "illustrates a display window containing program scheduling information" (column 4,

lines 50-51), which is merely a screen display of an EPG. The screen display is a rasterized representation of pixel information in a frame buffer. It does not actually include, for example, the title of a program, the program's running time, etc. Numerous data structures would doubtlessly be used to create this screen display, including, for example, files, lists, arrays, records, trees, tables, etc. However, Sullivan is completely silent about what specific data structures are used.

The cited passages of Sullivan refer to "program scheduling information." However, this does not refer to a specific type of data structure or give any insights into how it is organized. Rather, it refers to a kind of data, *i.e.*, data related to program scheduling. Sullivan could format this data in a variety of ways, but he chose to remain silent about specific data structures used to implement his EPG. Even if Sullivan's Figure 3A could be considered a representation of a data structure (which it cannot), Applicant respectfully points out that Sullivan's Figure 3A relates to multiple television programs (World News, Seinfield), not a <u>particular</u> television program, as claimed.

Similarly, Figure 3B does not illustrate a data structure, but, rather, a window for displaying program information. The window does not "store" the program attribute data, but is merely a mechanism for displaying them. For example, the program name of "Seinfeld" is stored somewhere in memory in a data structure, such as a table, array, etc. Each piece of program information could, and likely would, be stored within a separate database table. However, Sullivan is silent about how the data is stored or organized. To infer that Figure 3B sheds light on how Sullivan organizes program information in memory, or that all of the attributes are encapsulated within a single data structure, is manifestly unreasonable. This type of reasoning would be similar to looking

at a Windows XP desktop and inferring that all of the data represented by the folders, icons, and files contained therein are all represented within a single data structure. This is simply untrue.

Likewise, Figure 3C is being cited for the claimed user-selectable actions containing program code. In reality, Figure 3C is merely a window with some buttons, such as "Record" or "Remind." While program code must be located somewhere for carrying out these functions, there is nothing in Sullivan or the other cited references that teaches or suggests that the program code should be encapsulated in a PIO, along with the attribute data and graphical data. Any such inference must be impermissibly taken from the Applicant's own teachings.

2. The cited references do not disclose encapsulating attribute data, program code, and graphical data in a single data structure for a particular television program.

Even if Figure 3B depicted a data structure, which it does not, it only shows, at best, attribute data. Figure 3B does not, however, show program code for carrying out user-selectable actions and graphical data for showing a visual indicator (*e.g.*, an icon) representing the television program. Thus, Sullivan does not teach encapsulating attribute data with program code for user-selectable actions and graphical data for a visual indicator within the same data structure, as claimed.

Similarly, even if Figure 3C illustrated a data structure, which it does not, it only hints at the existence of user-selectable actions. Figure 3C does not show attribute data carrying information about a particular television program or graphical data for displaying a visual indicator. Thus, Sullivan does not teach encapsulating program code for user-selectable actions with attribute data and graphical data, as claimed.

Encapsulated means "encase[d] in or as if in a capsule" or that the PIO "form[s] a capsule or sheath around" the actions or attributes. See www.dictionary.com. None of the cited references show the forming of a capsule or sheath around the claimed attributes, actions, and visual indicator. Indeed, Sullivan teaches away from this interpretation by showing attributes and actions in separate windows (Figures 3B/3C).

3. The cited references do not disclose that the attribute data, program code, and graphical data associated with the particular program are transmittable ***as a unit*** from one interactive television system to another in response to the encapsulating PIO being sent between the interactive television systems.

One of the advantages of encapsulating the attribute data, program code, and graphical data within a single data structure is the ability to easily transmit that data structure from one interactive television system to another. This makes the PIO transportable, as depicted in Figure 15 of the present application.

None of the cited references teach or suggest transmitting a PIO between interactive television systems. Indeed, using the typical systems depicted, for example, in Sullivan and Lawler, the system would likely need to assemble information from a number of different data structures in order to transmit the equivalent of a PIO between interactive television systems. However, as described above, the prior art is silent about the specific data structures used for storing the recited data and program code. Indeed, the prior art is even silent about encapsulating the recited data and program code in a single data structure.

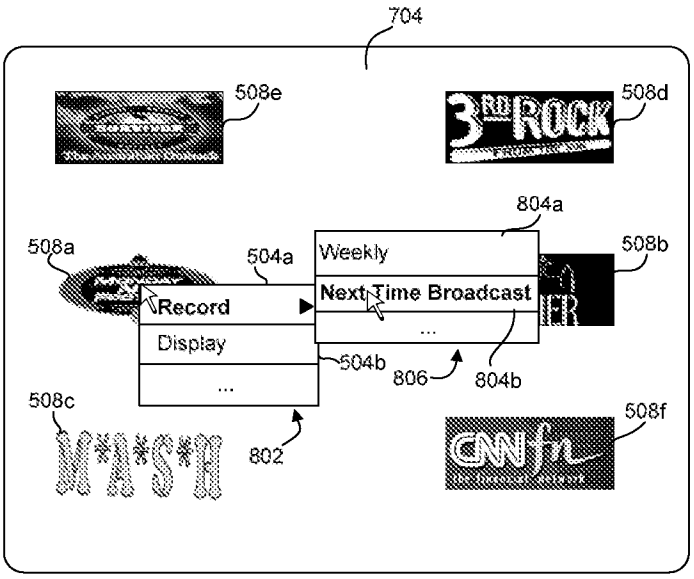
Maryka, cited by the Examiner in connection with claims 6, 28, and 52, merely discloses the existence of JavaBeans. Applicant never claimed to have invented JavaBeans, which is simply one technology for implementing the claimed PIOs. Maryka

uses JavaBeans to facilitate transfer of software programs between computers in such a way that a transfer interrupted by a power failure on the receiving device can be recovered. Maryka says nothing about using JavaBeans to encapsulate ***attribute data***, ***program code for user-selectable actions***, and ***graphical data for a visual indicator***, all pertaining to a ***particular television program***.

In view of the foregoing, Applicant respectfully submits that claims 1, 17, 41, and 65-67 are patentably distinct over the cited references. All other claims depend directly or indirectly on one of the foregoing claims and are therefore patentably distinct for at least the same reasons.

Claims 17, 64, and 67

As illustrated below in Figure 8 of the present application, the PIOs are not displayed in the traditional grid-type EPG format of Lawler, which consists of rows and columns corresponding to channels and time slots. The claimed visual indicators may be displayed individually or in combination with other visual indicators using a desktop metaphor. A user may click on, highlight, or otherwise select one of the visual indicators, which results in a context menu or the like being displayed. A user may then select an action from the context menu, which causes the associated code within the PIO to be executed by the interactive television system.



The Office Action states that Sullivan is "silent with [respect to] the graphical user interface is other than a grid-based electronic program guide with rows and columns corresponding to channels and timeslots." Far from being silent, Sullivan discloses a grid-based EPG that directly contradicts the claim limitations. For example, Figure 3A clearly shows Sullivan's grid-type interface.

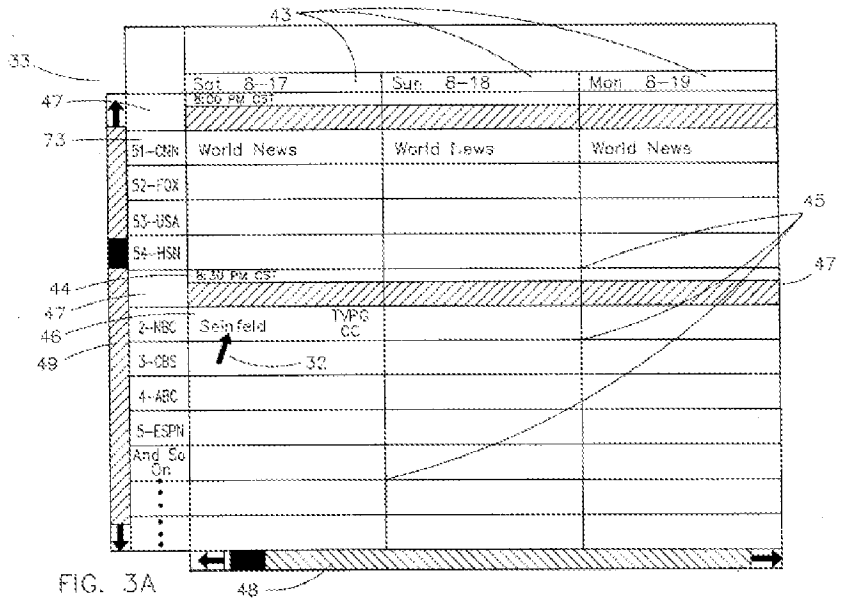


FIG. 3A

The Office Action refers to Niijima for a "graphical user interface [that] is other than a grid-based electronic program guide with rows and columns corresponding to channels and timeslots." While it is true that Nijiima's guide does not specifically refer to timeslots, it is nonetheless based on a grid and includes columns corresponding to channels. This still results in all of the shortcomings of grid-based guides mentioned in the present application:

> While EPGs have numerous advantages over conventional printed guides (such as TV Guide®), EPGs are still based on the channel/time slot model, which is of diminishing importance today.
>
> * * *
>
> Forcing a viewer to search through a grid consisting, for example, of over 500 rows (corresponding to channels) and possibly thousands of columns (corresponding to time slots) is no longer acceptable. Given the wide variety of entertainment options and the limited amount of time available to individuals for entertainment, any advancement increasing the convenience of an entertainment system would be highly advantageous.

Specification at page 2.

Applicant has amended claims 17, 64, and 67 to recite that the "the graphical user interface is other than a grid-based electronic program guide with rows corresponding to channels." This limitation is clearly contradicted by Niijima.

Even overlooking the fact that Sullivan and Niijima teach away from the claimed invention, the combination of cited references does not teach the other elements of claim 17. As amended, claim 17 recites the steps of:

> (1) displaying a plurality of visual indicators of respective PIOs;
> (2) receiving a user selection of a PIO through its visual indicator;
> (3) displaying a list of available actions for the selected PIO;
> (4) receiving a user selection of one of the available actions; and
> (5) executing the program code included with the PIO for the selected action within the interactive television system.

The cited references do not disclose or suggest displaying a plurality of visual indicators of respective PIOs, as shown in Figure 8 of the present application. Moreover, the cited references do not disclose displaying a list of actions in response to a user selection of one of the visual indicators. After a user has selected one of the available actions, none of the cited references disclose or suggest "executing the program code included within the PIO for the selected action within the interactive television system." As explained above, none of the cited references disclose or suggest that the program code for the selected action is actually stored within a PIO. Such a reading is simply not supported by Sullivan. Accordingly, claims 17, 64, and 67 are believed to be patentably distinct.

Claims 5, 27, and 51

Claims 5, 27, and 51 recite that:

> the program code is ***machine-independent format*** that is executable in a ***virtual machine*** within the entertainment device and ***any destination device to which the PIO is sent***, such that the ***program code does not need to be installed on the destination device*** prior to receiving the PIO in order to perform an associated user-selected action.

Support for this amendment may be found, for example, in pages 18-20 of the present application.

These claimed features allow a destination device to which a PIO is sent (*e.g.*, another set-top box) to execute the one or more actions associated with the PIO without requiring the program code to be pre-installed on the destination device. Furthermore, providing the program code in a machine-independent format allows PIOs to be shared

between a variety of different types of devices, such as cellular phones, personal computers, and set-top boxes (STBs).

As an example, a user may transmit a PIO representing a television program from an STB to her cellular telephone. The PIO may include an action for displaying one or more of the attributes of the PIO, *e.g.,* the starting time of the television program. The cell phone does not need to have software installed for examining the PIO and outputting the requested information. Instead, the action, as represented by machine-independent code within the PIO, may be executable by a virtual machine within the cellular telephone to output the attribute information, either on the telephone's display screen or to the user's personal information manager (PIM).

The term "machine-independent" is a term of art that is used to refer to program code that can run on different types of machines:

> In computer science, a machine-independent program is any program that can be run by any computer, without regard to its architecture or operating system. Any well-written Java or .NET application could be machine-independent because these platforms run on virtual machines on top of the real computer. The real machine-dependent part is the virtual machine, so this is the (usually little compared to the class libraries) chunk of code that needs to be ported. (http://en.wikipedia.org/wiki/Machine-independent).
>
> Computer hardware device, code, debugger, compiler or other form of software program or hardware device that requires little or no change to work with other hardware or computers. (http://www.computerhope.com/jargon/m/machinde.htm).

The Office Action appears to be using the term "machine-independent" in a completely different way, contrary to the meaning understood by a person of ordinary skill in the art. The Office Action refers to column 3, lines 56-67 of Sullivan for these limitations, which is reproduced below:

> The present invention further provides a signal embodied in a propagation medium in a system using program scheduling information. The signal comprises at least one instruction configured to select a program from the program scheduling information, at least one instruction configured to compare a program name for the selected program to the program scheduling information, and at least one instruction configured to determine whether to schedule at least one event for the program for at least one predetermined time slot based upon the results of comparing the program name for the selected program to the program scheduling information.

Applicant is unable to see how the foregoing passage from Sullivan is relevant to the claimed invention. It does not mention or even hint at machine-independent program code. It does not disclose or suggest a virtual machine. It does not disclose or suggest that a PIO may be sent to a destination device, allowing the destination device to enjoy the functionality of the PIO (user-selectable actions) without having the program code for those actions pre-installed on the destination device. Applicant respectfully submits that none of the limitations of claims 5, 27, and 51 are taught by Sullivan. Applicant, therefore, respectfully requests clarification from the Examiner.

The cited references do not disclose, individually or collectively, a PIO comprising program code "in a machine-independent format that is executable in a virtual machine within the entertainment device and any destination device to which the PIO is sent." Thus, none of the cited references can avoid having "the program code … installed on the destination device prior to receiving the PIO in order to perform an associated user-selected action." Accordingly, claims 5, 27, and 51 are believed to be patentably distinct over the prior art of record.

Claims 3, 25, and 49

As amended, claims 3, 25, and 49 recite that a PIO may not only encapsulate a plurality of attributes, actions, and visual indicators associated with a particular program, the PIO may also encapsulate an audible indicator capable of being played back by the interactive television system. As shown in Figure 5 of the present application, all of these elements are encapsulated within a single data structure. This simplifies sharing of content between multiple users. For instance, a user need only select a particular PIO by its visual indicator and send it to a friend using e-mail or another suitable means to allow the friend to view information about the program, execute actions with respect to the program (*e.g.*, record the program), and listen to audio content.

None of the cited references disclose encapsulating an audible indicator with attribute data, program code, and graphic data associated with a particular television program. The Office Action refers to Sullivan for this limitation. However, the passages referred to by the Office Action (column 9, lines 18-22; column 12, lines 58-64) merely state that Sullivan's receiver 2 is capable of outputting an audio signal. Sullivan says nothing about encapsulating audio data within a PIO. Applicant respectfully points out that a PIO is not a television program, but an object used to represent a television program within an electronic program guide. The claimed audible indicator is not the audio content of the television program that is output once a person tunes to the program. Rather, the audible indicator, like the visual indicator, is something to help a user find a desired program within the guide. It might include, for example, a pre-recorded clip from the television program, a recording of the spoken name of the program, etc., which could be used to help a visually-impaired person locate the desired program.

Conclusion

A rejection based on prior art – whether grounded in anticipation or obviousness – must account for each and every claim limitation. *Celeritas Techs. Inc. v. Rockwell Int'l Corp.*, 150 F.3d 1354, 1360, 47 U.S.P.Q.2d 1516, 1522 (Fed. Cir. 1998) (anticipation); *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q.2d 494, 496 (CCPA 1970) (obviousness); MPEP § 2143.03 ("To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art.") (emphasis added). In the present case, Applicant respectfully submits that none of the cited references disclose the unique structure of the PIOs recited in the independent claims.

For at least the foregoing reasons, the cited prior art references, whether considered individually or in combination, fail to disclose each of the limitations in any of the pending independent claims. For at least the same reasons, each of the claims depending therefrom are also patentably distinct from the cited prior art. Therefore, all claims are believed to be in condition for allowance. A Notice of Allowance is respectfully requested. The Examiner is encouraged to contact the undersigned at the telephone number provided below for a quick resolution of any remaining issues.

Respectfully submitted,

Digeo, Inc.

By /Kory D. Christensen/
Kory D. Christensen
Registration No. 43,548

STOEL RIVES LLP
One Utah Center Suite 1100
201 S Main Street
Salt Lake City, UT 84111-4904
Telephone: (801) 328-3131
Facsimile: (801) 578-6999